DOCUMENT RESUME

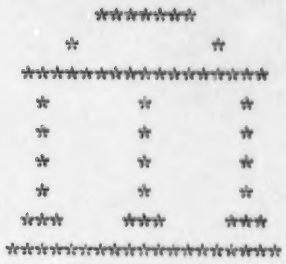
ED 247 929 IR 050 806

AUTHOR	Bolin, Robert L.
TITLE	FAMULUS: A Personal Information System User's Manual (for IBM System Users).
INSTITUTION	Georgia Univ., Athens.
PUB DATE	1 Aug 84
NOTE	66p.; For related document, see ED 053 754.
PUB TYPE	Guides - General (050) -- Reference Materials - General (130)
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	Computer Oriented Programs; *Computer Software; Guidelines; *Information Retrieval; *Information Storage; Information Systems; Input Output; Search Strategies
IDENTIFIERS	*FAMULUS; Job Control Language; University of Georgia

ABSTRACT

This manual explains how to use the FAMULUS program package in a computing environment where most text is input through terminals and most files are stored on disks. Developed in the late 1960s by the Pacific Southwest Forest and Range Experiment Station of the United States Forest Service, FAMULUS was conceived as a personal documentation system for research scientists. It provides extreme individual freedom to structure and update information files maintained by individuals, and interchange and merger of files between individuals or groups is easily accomplished. This manual was adapted from a FAMULUS manual prepared in 1969 and enhanced with some ideas and examples from three other FAMULUS manuals. The main change was to recommend using an interactive editor program instead of keypunching to prepare the FAMULUS records. The manual is divided into four chapters: (1) general information, (2) preparing an input file, (3) the FAMULUS programs, and (4) effective use of the search program. The four appendices include a bibliography and information on rules for keypunching (the original mode of input), how to interpret error messages, and job control language. (DMC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****



U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED247929

FAMULUS: A PERSONAL INFORMATION SYSTEM

USER'S MANUAL

(For IBM System Users)

Robert L. Bolin

Baldwin Data Analysis Center
University of Georgia
Athens, GA 30602

August 1, 1984

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Robert L. Bolin

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR050806

PREFACE

This manual explains how to use the FAMULUS program package in a computing environment where most text is input through terminals and most files are stored on disks. I adapted it from a manual prepared in 1969,[1] and I have gotten some ideas and examples from three other manuals.[2] The main change I made was to recommend using an interactive editor program to prepare and perfect the FAMULUS records. The original manual described this mode of operation:

1. Preparing records on punched cards.

2. Creating a master file on tape by using the FAMULUS program called EDIT.

3. Editing records or adding new ones to the tape master file file created in Step 2 by using the FAMULUS program EDIT.

I advocate:

1. Preparing records in an input file (on disk files) by using an interactive editor program.

2. Creating a FAMULUS master file, on disk or tape, by using the FAMULUS program called EDIT.

3. Editing or adding new records to the input file prepared in Step 1.

---

[1] FAMULUS: a personal documentation system . . . , User's Manual . . .
Berkeley, CA: US Forest Service, Pacific Southwest Forest and Range Experiment Station 1969.
Available from: NTIS, Springfield, VA; PB 202-534.
Also available from: ERIC Document Reproduction Service, Arlington, VA: ED 053-754.

[2] FAMULUS Reference Manual.
Corvallis, OR: Oregon State University, Milne Computer Center; 1981.

FAMULUS Reference Manual for the 1110.
Madison, WI: University of Wisconsin-Madison, Academic Computing Center; 1977.

FAMULUS Users Manual.
London: University College London, Computer Centre; 1972.

4. Using the FAMULUS EDIT program to replace the master file. This step is a repeat of Step 2.

Since the FAMULUS master file is never edited--only replaced--that mode of operation eliminates the need to use the editing capabilities of the FAMULUS EDIT program. Therefore, I eliminated the sections on the editing capabilities of EDIT from this manual. See Chapter 2, PREPARING AN INPUT FILE, for a description of the use of an interactive editor program to create an input file.

FAMULUS was developed in the late 1960s by the Pacific Southwest Forest and Range Experiment Station of the US Forest Service. The background of FAMULUS is explained in the foreword to the original manual:

> FAMULUS was conceived by Theodor B. Yerke, Station Librarian, as a personal documentation system for research scientists. The system provides extreme individual freedom to structure and update information files maintained by individuals. Interchange and merger of files between individuals or groups is also easily accomplished.

The PSW Forest and Range Experiment Station provided the FAMULUS software to the University of Georgia free of charge. The Political Science Department has supported the effort to obtain and implement FAMULUS at UGA. Bill Blum of the Office of Computing and Information Services, John Alford and Sylvia Jensen of the Political Science Department, and Judy Ishee and Larry Tipton of the Small Business Development Center assisted with that effort. Tom Deitz of the University Libraries read this manual carefully and suggested many improvements. Doug McLeod of the Sociology Department installed the load module to make the FAMULUS programs available.

Bob Bolin
Graduate Student

CONTENTS

PREFACE i

Chapter

page

1. GENERAL INFORMATION 1

Applications 1
The FAMULUS Programs 2
The FAMULUS Master File 3

2. PREPARING AN INPUT FILE 4

Record Design 5
Record Formats 6
Field Labels 6
Testing 7
Selecting Indexing Terms 8
A Controlled Vocabulary 10
Quality Control 11
The Role of an Editor in Quality Control 11
Using Searching for Quality Control 11
Using Sorting for Quality Control 12

3. THE FAMULUS PROGRAMS 14

Basic Concepts and Terms 14
A Job 14
Getting Help When FAMULUS Jobs Fail 15
Input and Master Files 16
Recommended Mode of Operation 16
The EDIT Program 17
/ID/ Card (required) 18
/FIELDS/ Card (required) 18
/DESCRIPTOR FIELD/ Card (optional) 19
/ORIGINAL/ Card (required) 19
/CITATIONS/ Card (required) 19
The SORT Program 20
/ID/ Card (required) 20
/FIELDS/ Card (required) 20
/SELECT/ Card (optional) 21
The MERGE Program 21
/ID/ Cards (required) 22
/NEW ID/ Card (optional) 22

/PRINT/ Card (optional) 22
The OSSIFY Program 23
/ID/ Card (required) 23
/SELECT/ (Optional) 23
The GALLEY Program 24
/ID/ Card (required) 24
/PRINT BY FIELDS/ Card (optional) 24
/SELECT/ Card (optional) 24
/WIDTH/ Card (optional) 25
The VOCAB Program 25
/ID/ Card (required) 26
/FIELDS/ Card (optional) 26
/SELECT/ Card (optional) 27
/VOCABULARY/ Card (required) 27
Statistics Produced 27
The INDEX Program 28
/ID/ Card (required) 28
/SELECT/ Card (optional) 28
/PUNCH/ Card (optional) 29
The SEARCH Program 29
/ID/ Card (required) 30
/FIELDS/ Card (optional) 30
/WRITE TAPE/ Card (optional) 30
/SEARCH/ Card (required) 30

4. EFFECTIVE USE OF THE SEARCH PROGRAM 33

Conjunction 33
Disjunction 34
Constructing the Search Formula 35
Formula Structures 36
Examples of Searches 37

Appendix

page

A. RULES FOR KEYPUNCHING 38

PROGRAM CONTROL CARDS 38
RECORDS 38

B. HOW TO INTERPRET ERROR MESSAGES 41

C. JOB CONTROL LANGUAGE 43

The FAMULUS Job 44
The JOB Card 44
The JES3 Statements 44
The Steps 44
The Step Card 45
DD Statements and Unit Numbers 45
Unit Numbers Used by FAMULUS Jobs 48

JCL Samples 49
EDIT JOB 49
SORT and GALLEY JOB 50
MERGE JOB 51
OSSIFY Job 51
GALLEY Job--to write to the laser printer 52
VOCAB Job 53
INDEX Job 53
SEARCH Job 54
DS DUMP Job--to write a file on a tape 54
DSDUMP Job--to create an IBM file 55
SORTD Job 55

D. BIBLIOGRAPHY 56

REFERENCES 56
FAMULUS Articles 58

Chapter 1

GENERAL INFORMATION

1.1 APPLICATIONS

FAMULUS is a package of computer programs which allows the user to build, merge, sort, search, index, and list computer files. It gives the individual user powerful information storage and retrieval capabilities.

Although it is powerful, FAMULUS is simple and flexible. It has been widely used by researchers and librarians. Some uses are described in the FAMULUS Articles section in the BIBLIOGRAPHY.

In addition to its bibliographic applications, FAMULUS can be used in instruction. For example:

1. Principles of indexing and abstracting could be taught by having students create files and then search them.

2. The concept of computer files and the use of IBM Job Control Language could be taught by having students use FAMULUS to create, move, sort, search, and list files.

3. Boolean logic could be demonstrated by having students search a sample file using various search formulas.

Almost any small file--the sort traditionally kept on index cards in a shoe box--can be replaced with a FAMULUS computer file. Here are some indexes that could be prepared easily using FAMULUS:

1. People can be matched with their phone numbers.

2. Artifacts with archeological sites.

3. Computer software with its applications.

4. Gardeners with their plots.

5. Terms with their definitions.

6. Plans with buildings.

7. Artists with slides showing their works.

Of course, the big advantage of a computer file over a shoe-box file is that the computer file can be sorted, listed, and copied with ease. For example, an "expert file" listing faculty members and their research interests could be prepared using FAMULUS. Listings sorted by name, by department, and by field could be printed easily. In addition, the FAMULUS file could be searched to find faculty members with particular combinations of interests.

1.2 THE FAMULUS PROGRAMS

There are eight separate FAMULUS programs. The first four listed below are files management programs. The first three create FAMULUS master files. Those master files can be "read" only by the FAMULUS programs.[3] The fourth translates the contents of a master file into readable form and writes it to cards or a computer file.

1. EDIT--creates a FAMULUS master files using records supplied by the user. Those records are read from an input file. The input file may be a deck of punched cards or a computer file.

2. SORT--reads a FAMULUS master file, sorts its contents, and creates a second master file--a reorganized master file.

3. MERGE--combines two master files to create a third.

4. OSSIFY--writes out the contents of a master file in a readable format--the reverse of the action of the EDIT program.

The remaining four programs write reports. They list the contents of a FAMULUS master file or extract information from it.

5. GALLEY--produces a printout listing the contents of a master file.

6. VOCAB--prints, in alphabetical order, the words found in specified fields in the records in a master file.

7. INDEX--produces an index to a master file showing the records containing each descriptor (term in the designated descriptor field).

---

[3] FAMULUS master files are stored in a special format so that they cannot be read by ordinary means. You cannot display the contents of a FAMULUS master file at a computer terminal and make any sense of it.

8. SEARCH--produces a printout listing records satisfying search specifications defined by the user. Complex searches using Boolean logic--searches using terms connected by and, or, or not --can be performed.
SEARCH can also be used to create a new master file containing a subset of the file searched. It does that by writing the results of a search into a file.

1.3 THE FAMULUS MASTER FILE

The FAMULUS master file is the key to the system--it is your "database." All of the FAMULUS programs--except EDIT--read the master file and act on its contents. Some of the programs create new master files and others produce listings and reports.

The Most Commonly Used FAMULUS Programs

The typical FAMULUS project requires the use of only four programs:

1. EDIT to create the initial master files.
2. SORT to create a couple of reorganized master files.
3. GALLEY to make printouts from the master files.
4. And SEARCH to printout selected portions of the master file.

The other FAMULUS programs will be used less often, but they can prove handy.

Chapter 2

PREPARING AN INPUT FILE

The input file processed with the EDIT program should contain high qualtity records. The basic unit of a FAMULUS master file is the "record." A record is analogous to an individual index card in a shoe-box file. A FAMULUS record may be up to 4,000 characters long and may contain as many as 10 separate fields of information. In a file used to store a bibliography, the records might contain fields for the author's name, the title, the publisher, the publication date, keywords, and an abstract. Records in a dictionary file might contain fields for the term, definition, reference to related terms, the source of the information, and comments. Personnel record might contain fields for the worker's name, social security number, position, paygrade, date of appointment, and comments.

The FAMULUS program called EDIT reads the input file supplied by the user and writes the records found in the input file into a FAMULUS master file. The FAMULUS master file is the "database" which is read by the other FAMULUS programs.

As with other computer systems, the results produced by FAMULUS are only as good as the records supplied. Because the FAMULUS user controls the project from top to bottom and usually does not rely on records supplied by others, he or she has an advantage over most other computer users. The FAMULUS user designs the records, selects the records to input, oversees creation of the computer master file, and orders tailor-made printouts. This chapter is intended to help the user:

1. Design an appropriate record,
2. Select a suitable indexing scheme,and
3. Assure that only consistent and accurate records are included in the input file.

2.1 RECORD DESIGN

Before you can start a FAMULUS indexing project, you must make some basic decisions. Those include:

How the records are to be formatted, and

How many fields are needed.

Those questions are important because they determine what the access points for the records will be. For example, if you are creating a bibliographic file you would probably record at least three items of information:

1. Author name.

2. Title.

3. Journal citation or the facts of publication.

All three of those items of information could be recorded in a single field.

If that approach were used, the citation for Kate Turabian's Manual for Writers would appear on two lines as shown below.[4]

```
ITEM TURABIAN, KATE L. A MANUAL FOR WRITERS OF . . .
     TIONS. 4TH ED.: THE UNIVERSITY OF CHICAGO . . .
```

That would be satisfactory if you only want to produce a bibliography sorted by the author's last name. You could not prepare a listing sorted by title.

If you want title listings as well, at least two fields would be needed, e.g.

```
AUTH TURABIAN, KATE L.
CITA A MANUAL FOR WRITERS OF TERM PAPERS, THESES, . . .
      UNIVERSITY OF CHICAGO PRESS: 1973.
```

To create a title listing, you would use the SORT program to create a master file in which the title field comes first in each record and then use GALLEY to list the contents of the sorted master file.

---

[4] In the example, the second line is a continuation of the first. "TIONS" is a continuation of the word "dissertation." "Dissertation" was broken between lines because of the FAMULUS keypunching rules. Those rules are described succinctly in Appendix A, RULES FOR KEYPUNCHING.

If you want to be able to arrange the citations by date of publication or all citations for each author by date, three fields will be needed, e.g.

AUTH TURABIAN, KATE L.
CITA A MANUAL FOR WRITERS OF TERM PAPERS, THESES, . . .
UNIVERSITY OF CHICAGO PRESS.
DATE 1973.

2.1.1 Record Formats

The first step in designing a FAMULUS record is to select a record format--the basic information elements in the records and the order in which they will appear. A good practice is to pattern your FAMULUS records after some standard for bibliographic references. One standard format for references widely used for theses and dissertations is described by Turabian.[5] In her preface, Turabian lists the standards for references in several disciplines--physics, biology, mathematics, and psychology.

A good general-purpose format is the American National Standard for Bibliographic References. That standard is described briefly by Ellis Mount.[6]

2.1.2 Field Labels

After selecting a record format, the next step is to decide how the record will be divided into fields and how the fields will be labeled.

The information elements in a record format need not correspond one for one with the fields in the FAMULUS record. A single element in the record format could be divided into several fields. For example, your record format might include "authors" as an information element, but you might choose to use separate fields for first, second, and third authors. Also, several elements in the record format may be recorded in a single field. The format for a personnel record might have date of birth, sex, and veteran status as separate information

---

[5] Turabian, Kate L. A manual for writers of term papers, theses, and dissertations. 4th ed. Chicago: The University of Chicago Press; 1973. 216p.

[6] Mount, Ellis. A national standard for bibliographic references. Journal of the American Society for Information Science. 28(1): 3-12. 1977.

elements, but you could choose to include that information in a single field.

The fields should correspond to the access points you feel you may need. Here is an absurd case. You could have separate fields for the author's first, middle, and last names. You could produce listings showing all authors with the first name "ROBERT" or the middle name "LUKE." But why would you want to? Similarly, you could have separate fields for the place of publication and for the publisher's name. However, a single field for the facts of publication (the journal name or the place of publication and the publisher's name) would probably be sufficient.

Labels for the fields may be one to four characters long, must begin with an alphabetic character, and may not include special characters such as punctuation.

1. 1AUT (not correct)
2. AUTH (legal field label)
3. PUB. (not correct)
4. PUBL (legal field label)
5. NOTES (not correct)
6. N1 (legal field label)

Labels should be easy to type and distinctive. A2ZW would be a poor label because it is hard to type. Similarly, FEL1, FEL2, FEL3, FEL4, and FEL5 would be poor labels because they are distinguished only by the digit at the end. Labels like AUTH, TITL, PUBL, DATE, and NOTE are distinctive and easy to type.

2.1.3 Testing

Once you have decided on a record structure and on your field labels, you should test your records. Try writing out some typical records in input format (See Appendix A, RULES FOR KEYPUNCHING). If the records look proper, then try creating a small FAMULUS master file using some representative records. Try running different FAMULUS programs--especially the SORT program--using that master file. If you are satisfied with the printouts and did not have trouble entering the records, you are ready to use your records on a larger scale.

2.2 SELECTING INDEXING TERMS

The FAMULUS program named INDEX produces an index to a FAMULUS master file. The index is an alphabetical listing of the terms found in the descriptor field[7] in the records in the file. The reference numbers of the records in which the term was found appear below the term in the listing. Here is what an index printout for a small test file looks like:

```
      I N D E X

CATALOGING OF MANUSCRIPTS
     19
CATALOGING OF RARE BOOKS
     19
ECOLOGY--BIBLIOGRAPHY
     6
ELECTRONIC DATA PROCESSING--LIBRARY CATAL
     8
HUMANITIES--DATA PROCESSING
     16
INDEXING--STUDY AND TEACHING
     20
INFORMATION STORAGE AND RETRIEVAL SYSTEM
     12,16,27
LINGUISTICS--DATA PROCESSING
     27
MISTLETOE--BIBLIOGRAPHY
     24
VIRUS DISEASES OF INSECTS--CATALOGS AND
     18
```

The numbers in the index refer to the sequence of the records in the FAMULUS master file. You can find out what the order of the records is by running the GALLEY program. GALLEY prints the records in order with their sequence numbers to the right. The sixth entry in the listing for this master file is on ECOLOGY:

---

[7] The descriptor field is specified by using a /DESCRIPTOR/ control card when the EDIT program is run to create the master file. The use of the /DESCRIPTOR/ card is described in Section 3.2.3.

6. COCHRAN, M.I.; THOMAS, J.M.. APPLICATION OF FAMULUS: A BIBLIOGRAPHY OF QUANTITATIVE ECOLOGY. REDLAND, WA: BATTELLE PACIFIC NORTHWEST LABS; 1976. AVAILABLE FROM: NTIS, SPRINGFIELD, VA; BNWL-2130. DISTRIBUTED BY ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION TO DEPOSITORY LIBRARIES FOR ERDA MATERIAL. ECOLOGY--BIBLIOGRAPHY.

The eighth is on ELECTRONIC DATA PROCESSING:

8. CRENNELL, K.M.; MARSH, E. COMPUTERISATION OF THE RUTHERFORD LABORATORY CATALOGUE. PROGRAM. 13(2):73-84; 1979 APRIL. ELECTRONIC DATA PROCESSING--LIBRARY CATALOGS

ECOLOGY-BIBLIOGRAPHY and ELECTRONIC DATA PROCESSING--LIBRARY CATALOGS appear in the index because those sets of indexing terms were used in the descriptor fields in Records 6 and 8.

A descriptor field need not be used in FAMULUS records, but one must be used if the INDEX program is to be used. For the INDEX program to work properly the indexing terms used in the descriptor field in the records must be standardized and consistent.

Obviously, the terms appearing in an index must be used consistently. If "dog," "house pet," "K-9," "mans' best friend," and "canine" had been used to index different articles referring to the same sort of beast, the index in which those terms appeared will not be very useful to the person who wanted to find all the articles on dogs quickly. However, if "dog" were used to index each article related to dogs and the searcher knows that, the index would be very useful. Consistent use of indexing terms can be assured by using a subject headings list which shows the prefered term to represent each concept. The searcher who uses a subject headings list is said to be using a "controlled vocabulary." If a controlled vocabulary is used to index a file, a searcher may use the subject headings list to find the correct term to use when searching for references to a given concept.

The FAMULUS program named SEARCH can be used to search any or all fields in FAMULUS records. SEARCH lists all records containing or not containing a particular term or combination of terms in the specified fields. The SEARCH program can be used to search fields containing natural language vocabulary such as title fields--where the words which appear were supplied by the authors. If a field containing natural language vocabulary is being searched, the searcher must figure out what terms to search for. For example, he or she might need to make a search for all occurances of "dog," "K-9," or "canine" to find all the articles on dogs.

In contrast, searching a descriptor field where terms from a controlled vocabulary appear is much simpler. The searcher need only look up the terms to search for in the subject headings list. Searches using controlled vocabularies, using natural language terms, and using a combination of the two are discussed in detail in Chapter 4.

2.2.1 A Controlled Vocabulary

A particular controlled vocabulary is described in a guide--called a subject headings list or a thsaurus--showing which terms are to be used to index given concepts. Subject headings lists and thesauri show what terms to use and what not to use and give references from those not to be used to the preferred terms. Thesauri often have sections showing various relationships between terms. The Library of Congress Subject Headings and the Thesaurus of ERIC Descriptors are good examples of vocabulary controlling guides.

You could develop a subject headings list from scratch, but to develop a thorough list in any given field of knowledge is costly and time consuming. You would be well advised to find a suitable thesaurus or set of thesauri to use. The original version of this manual was written for researchers working for the Forest Service. It advised using four thesauri for indexing forest-related research collections:

1. A thesaurus of engineering and scientific terms.
2. A subject headings list for biological and agricultural terms.
3. A list of forestry indexing terms.
4. A water resources thesaurus.

Of course, the thesauri suitable for indexing forestry articles whould not be useful to someone in political science or drama. You must find the particular thesauri best suited for indexing your material.

Many thesauri are available. A good place to look for them is in the subject catalog of the library. In the University of Georgia Libraries, they are listed under the headings "THESAURI" and "SUBJECT HEADINGS."

If the thesauri available do not allow indexing with sufficient precision, you can supplement the thesaurus terms with your own key words. If you use keywords, you should keep a list of them and use them consistently.

You should read the sections on the EDIT, INDEX, and SEARCH programs in Chapter 3 and Chapter 4 before deciding to use the indexing capability of FAMULUS or selecting thesauri to use.

2.3 QUALITY CONTROL

The key to quality control is consistency. Consistency is even more important than strict accuracy. You should not hesitate to change the punctuation in a title or to make minor changes in the wording of a title or some other field to cause related records to fall out together when they are sorted. For example, in a collection of reports, some may begin with numbers like "THIRD ANNUAL REPORT..." If you want the reports to appear together in a title listing, you should drop the ordinal numbers from the titles. If literal transcription of the text is essential, you can state in a note that changes were made for the sake of consistency and then quote the information exactly as it appeared.

Use of a single person as an editor, of computer searching and editing capabilities, and of sorting all can help you maintain consistency.

2.3.1 The Role of an Editor in Quality Control

Ideally, a single, knowledgeable person should edit all records input. Often the best editor is the person who is supervising the project and who designed the records being used. The editor should review all records for accuracy and consistency. If the editor is also the input operator, he or she will probably catch many errors in the course of entering the records. As the project progresses, FAMULUS printouts listing the records processed previously can help the editor to maintain consistency.

2.3.2 Using Searching for Quality Control

A FAMULUS input file contains the records to be processed by the FAMULUS EDIT program. An input file can be produced by punching the records onto cards. However, if the input file is created in a computer file, the interactive "editor" program used to create the file can be used to check for consistency among records. An interactive editor program allows a user at a computer terminal to create, review, and change the information in a computer file.

An interactive editor program should not be confused with the FAMULUS program named EDIT. The FAMULUS EDIT program reads an input file and creates a FAMULUS master file which can be read only by one of the other FAMULUS programs. An interactive editor program is a feature of an interactive computing system--like the TSO and MUSIC systems available at UGA--which allows the user at a terminal to manipulate a computer file. I recommend using an interactive editor program when preparing an input file. The interactive computing system used with the

large IBM computer on campus is called TSO (for Time Sharing Option). It has a handy interactive editor called QED. The MUSIC (McGill University System for Interactive Computing) system also has an interactive editor program. The files created using the MUSIC interactive editor can be "dumped" into the IBM system for execution. Manuals on the TSO and MUSIC systems are listed in the REFERENCES section at the end of the manual.

The input operator--the person using the interactive editor program to create an input file--can use the searching capabilities of the interactive editor program to check for consistency in the file. For example, if a title contains the words BOY'S CLUBS the input operator might search for all occurences of BOY or CLUB in the records input previously. If the words appeared as BOYS CLUB in all similar entries, that usage probably should be adopted. Obviously, the person editing the records must trust the input operator before he or she will allow that type of editing by the operator. That is another good reason for the editor to enter the records.

The interactive editor program that is used to create the input file can be used to make corrections and changes in the input file. Changes may be needed because errors are found in the records. The records may contain typographical errors or syntactical errors--format errors which will be flagged on the printout produced by the EDIT program. Also, changes may be needed because the person editing the records can have second thoughts about some part of the input file. The editor might decide that the names of corporate authors should be rotated--COLLEGE OF AGRICULTURE should be AGRICULTURE, COLLEGE OF, for example. That type of global or universal correction can be made easily with an interactive editor program. In fact, it can be made TOO EASILY. You should exercise caution when making global changes because you can change parts of the record you do not want to change. A good practice is to make a search through the file for the item you intend to change before making the global change.

2.3.3 Using Sorting for Quality Control

Sorting the input file can help in spotting inconsistencies.[8] The result of sorting an input file is a listing in which all occurences of like fields appear together. Also all blank lines in the input file will be placed together. For example, if you had an input file containing records with AUTH, TITL, PUBL, and SUBJ fields and the file contained

---

[8] The SORTD utility program is handy for sorting files. That utility is described in Brown (System/370, Job Control Language, Pages 220-224) and samples of the job control language needed to execute SORTD are given in the FACILITIES AND ACCESS SERVICES GUIDE and in Section D.2 JCL SAMPLES.

about 200 records, a printout showing the contents sorted in ascending[9] order will contain 4 or so blank pages, several pages of continuation lines (which begin with 5 blank spaces because of the FAMULUS key punching conventions), a listing of all the lines beginning with AUTH, a listing of all beginning with PUBL, a listing of all beginning with SUBJ, and a listing of all beginning with TITL. The list begins with several pages of blanks because blanks are used to separate the records in a FAMULUS input file and blanks sort ahead of everything else.

Sorting is useful because like items appear together in sorted lists and differences stand out. For example, if records contain dates, a date with the letter O in place of the number 0 will fall at the top or bottom of the group of numbers in the sorted listing, depending on how you sort the file. Here is what part of a sorted list could look like:

```
DATE 1907.
DATE 1908.
DATE 1908.
DATE 1909.
DATE 1900.
DATE 1910.
DATE 1911.
```

The "1900" appears after "1909" in this list which is sorted in ascending order because the letter "O" is considered larger than the "9" in 1909 and the zero is smaller than the second "1" in 1910. Likewise, a date erroneously punched "1907" would appear after "1999" and before "2000".

Misspelled field labels or inconsistencies in abbreviations for journal titles or in the use of names and initials of authors are often apparent in sorted lists. Sorting may also help you spot missing items. For example, if you are indexing items that have sequential serial numbers, missing numbers will be obvious in a sorted listing of the field containing the serial number. Such a list sorted in ascending order might look like this:

```
F9SN  989.
F9SN  990.
F9SN  991.
F9SN  992.
F9SN  993.
F9SN  996.
F9SN  997.
F9SN  999.
F9SN 1000.
F9SN 1001.
```

---

[9] Ascending order is conventional alphabetical order.

Chapter 3

THE FAMULUS PROGRAMS

3.1 BASIC CONCEPTS AND TERMS

3.1.1 A Job

FAMULUS runs on the IBM main-frame computer. Each FAMULUS program is executed by submitting a "job" to the computer.

A job is a number of lines of text which tells the computer what to do and which may supply data to be used by the computer program. All FAMULUS jobs are made up of punched, 80-column cards or card-images. Cards are submitted to the computer through a card-reader machine and card images are created in a computer file and submitted by giving the computer a command which causes it to simulate the action of a card reader. Whether cards or card-images are used, the format of the instructions and text on each line is the same. Only the keypunch character set (which has no lower-case letters) may be used. Those characters are found on the keyboard of an IBM Model 29 Card Punch Machine.

A job consists of a series of system control cards, program control cards, and optionally, data. System control cards tell the computer:

1. What program to execute.
2. What files to read and write.
3. What equipment--card readers, disk drives, tape drives, etc.--to use.
4. How to do the job.

System control cards are in a peculiar language called Job Control Language (JCL, for short). The basic JCL needed to run the FAMULUS programs is described in Appendix D, JOB CONTROL LANGUAGE.

The JCL is followed by the program control cards which tell the particular program being executed what to do. This chapter describes the control cards used by each program. The headings in the table of contents for this chapter show the control cards used by each.

The input file--the data to be processed--which follows the control cards in a job is described in Chapter 2, PREPARING AN INPUT FILE. In this manual, preparing FAMULUS jobs for inputting is referred to as "keypunching" whether the job is prepared by punching cards or by creating a computer file. In either case the same rules apply.

3.1.2 Getting Help When FAMULUS Jobs Fail

FAMULUS is a set of computer programs written in the FORTRAN programming language. A FAMULUS program is executed by submitting a job, as discussed above. If a job fails to execute the program, the problem is usually with the Job Control Language (JCL). The JCL must be correct or the computer cannot know what to do.

Many people on campus are familiar with JCL. They include consultants in the User Services office of OCIS, programmers working for various departments, and researchers familiar with similar packages like the Statistical Package for the Social Sciences. However, most of them are not familiar with FAMULUS. So to get help when a FAMULUS job fails, you must frame your question properly. You definitely should not say:

Do you know anything about FAMULUS?
or
My FAMULUS job failed, can you help me?

Most people will say, "I don't know anything about FAMULUS."

You should avoid mentioning FAMULUS at all. Show your printout to the person you want to get help from and say something like this:

My job failed, what is wrong with my JCL?
or
What does "Completion Code--System 722" mean? How can I fix my JCL?

If the person helping you asks what program you are using, tell him or her that you are using FAMULUS which consists of a set of FORTRAN programs like SPSS.

3.1.3 Input and Master Files

The original FAMULUS manual, on which this manual is based, did not refer to "input files" or "master files" but, to "card decks" and "tapes." The user today has more options than the users of the original manual. The user need not submit the input file in the form of cards and have the computer write the master file on tape. I assume that the user will create FAMULUS jobs in computer files and will have the computer create the master file in computer files as well. I have adopted the more general terms to refer to files--"input file" and "master file."

3.1.4 Recommended Mode of Operation

As I have noted, FAMULUS was designed for use with card input and with output on magnetic tape. Feeding large card decks through a card reader was a time consuming, tedious, and expensive process. So that the user could avoid feeding decks over and over, the EDIT program was given the capability of editing master files. That allowed the user to perfect--eliminate errors in--the master file by changing or adding records without feeding in the whole card deck again. The OSSIFY program was provided to dump the perfected master file out as a deck of cards in the input format.[10]

The mode of operation I recommend using consists of:

1. Creating the input file in a computer file using an interactive editor program.

2. Using the FAMULUS EDIT program to create a FAMULUS master file.

3. Using the interactive editor program to edit and update the input file as needed.

---

[10] The user today does not need to handle cards. The input deck can be in the form of a file in the computer. An input file can be submitted simply by giving the computer a single command. I assume that the user of this manual will perfect the input file and use the EDIT program in the FAMULUS package only to create updated master files--not to edit master files. I have not included information about the editing capabilities of the EDIT program in this manual. Users who want to use EDIT to edit a master file should consult the original version of this manual. A copy is available in the University of Georgia Main Library in the ERIC microfiche collection under ED 053-754.

4. Using the FAMULUS EDIT program to replace the FAMULUS master file whenever the input file has been updated.

It is much easier to edit an input file directly using an interactive editor than to edit a FAMULUS master file using successive runs of the EDIT program. Input files can be created and edited easily using MUSIC (McGill University System for Interactive Computing), TSO (Time Sharing Option), or using a microcomputer with the capability to communicate with the IBM main-frame computer.

The MUSIC system is linked to the IBM system so that jobs prepared in a MUSIC file may be submitted to the IBM system. The job submitted to the IBM system can be an EDIT program to create a new master file, some other FAMULUS program, or some utility program (like DSDUMP which can be used to copy the input file to tape or disk files). MUSIC is simple to use and undergraduates can get access to the system easily.

The user will have to get access to the IBM system to run FAMULUS jobs. The user can use the TSO system available to anyone with access to the IBM system to create, edit, and submit FAMULUS jobs.

A microcomputer with communications capability can be used to submit a FAMULUS job. The job can be created and edited in a file on the microcomputer and then submitted to the IBM main-frame computer for execution. A microcomputer can also be used to "download" or "upload" files--that is, to copy them from or to the main-frame computer.

3.2 THE EDIT PROGRAM

EDIT creates a master file. It reads an input file, writes error-free records into the master file, and writes a printout listing all records read. It does not write records containing format errors into the master file but does write them on the print out, set off by asterisks and accompanied by an error message (See Appendix B, HOW TO INTERPRET ERROR MESSAGES).

Before the computer reads any lines of text, it reads the system and program control cards which must be at the head of the input file. Control cards are lines of text which tell the computer and the FAMULUS program what to do. The program control cards provide the name of the file, the list of field labels, and the name of the descriptor field. This information is written at the head of the master file created. Subsequent use of the master file requires that the /ID/ card supplied in the program control cards match the /ID/ information at the head of the master file being used.

The EDIT program accepts the following control cards (parameter "f" stands for "field label"):

```
/ID/file name              (required)
/FIELDS/(f1,f2,...,f10)    (required)
/DESCRIPTOR FIELD/(f1)     (optional)
/ORIGINAL/                 (required)
/CITATIONS/                (required)
```

3.2.1 /ID/ Card (required)

Punching begins in Column 1 with the control statement /ID/. The file name follows the second slash and may extend to Column 80. The name may be continued onto the next line. If it is continued, the continuation should begin in Column 1 of the second line. The file name can be up to 100 characters long. Blanks before and after the file name do not count. All subsequent use of any master file by any FAMULUS program requires an /ID/ card which exactly matches the identification written at the head of the master file. In every FAMULUS job submitted, the /ID/ card must be the first program control card. Some examples:

```
/ID/LAB NOTES--PATHOLOGY PROJECT
/ID/PSW F&R EXPERIMENT STATION LIBRARY CATALOG
/ID/UNIV. OF GEORGIA BULLETIN.
```

3.2.2 /FIELDS/ Card (required)

This program gives the names, order, and number of the field labels. You may have up to 10 fields, naming them as you wish.[11] Each label must begin with an alphabetic character and may be up to four characters long. The other characters may be either letters or numbers, but not special characters such as punctuation. Two fields may not have the same name.

The /FIELDS/ card sets the order in which the information in a record will appear in the FAMULUS master file created. If the records input contain field labels not in the same order as called for by the /FIELDS/ card, the EDIT program will assume the cards are in order. It will skip them and print an appropriate error message. Here is what /FIELDS/ cards look like:

```
/FIELDS/(AUTH,TITL,DATE)
/FIELDS/(F1YR,VLNU,ONAM,TITL,SNAM,SNUM,UNAM,UTIL,F9SN,NOTE)
/FIELDS/(AUTH,TITL,IMPR,AVAI,NOTE,XREF,SUBJ,F9,F10)
```

---

[11] Chapter 2, Section 2.1.3 Field Labels gives some guidelines for selecting labels.

3.2.3 /DESCRIPTOR FIELD/ Card (optional)

One of your fields may be designated as the descriptor field.[12] FAMULUS programs regard the contents of a descriptor field differently from the contents of other fields. In every other field words are recognized as such when they are separated by blanks. But in a declared descriptor field the program does not look for words separated by blanks. It searches for units separated by commas. This makes possible the inclusion of index terms consisting of multi-word phrases.

A descriptor field containing of the terms SAWLOGS, BLUE SAP STAIN, and SCALING will be regarded by the INDEX, VOCAB, and SEARCH programs as having three recognizable terms. This presents a problem since the phrase BLUE SAP STAIN in a descriptor field is handled as though it were one large word. If you were using the SEARCH program with a descriptor "SAP STAIN", it would not recognize "SAP STAIN" in the phrase "BLUE SAP STAIN," and you would not retrieve this term. In the other fields--those not designated as the DESCRIPTOR field--the SEARCH program can recognize the term "SAP STAIN" in the phrase "BLUE SAP STAIN."

Only one field label may appear on a /DESCRIPTOR FIELD/ card, like:

```
/DESCRIPTOR FIELD/(SUBJ)
```

3.2.4 /ORIGINAL/ Card (required)

This card is required by the EDIT program.

3.2.5 /CITATIONS/ Card (required)

This card always must be the last program control statement in an EDIT job and must be followed immediately by the input file containing the records to be processed.

---

[12] The use of descriptors is described in detail in Chapter 2, Section 2.2 Selecting Descriptor Terms.

3.3 THE SORT PROGRAM

The SORT program creates a new FAMULUS master file with the records sorted in the order you prescribe. The most common use of SORT is to obtain an alphabetical listing of bibliographic records by author or title. All of the allowable ten fields may be reordered if desired, or a simple alphabetical listing using only one field as the key can be made. The order in which you want the records is shown on the /FIELDS/ card. If only an alphabetical listing by author is desired, then the /FIELDS/ card might be: /FIELDS/(AUTH,TITL). The second field label is added so that in the event that two identical authors are encountered, the SORT program can use the titles to establish precedence. Any fields not specified on the /FIELDS/ card will remain in the order of the master file being read by the sort program.

Output for the SORT program consists of a listing of the program control cards, and some technical information about the number of groups of records which were processed. This is followed by three large "OK's" if the processing was completed successfully. A listing of the sorted records is not printed. GALLEY, the FAMULUS printing program, can be used to print the newly sorted file.

The SORT program accepts the following control cards (parameter "f" represents "field label"; "n" represents "record number"):

```
/ID/file name            (required)
/FIELDS/(f1,f2,...f10)   (required)
/SELECT/(n1,n2,...)      (optional)
```

3.3.1 /ID/ Card (required)

The /ID/ card must match the file name written at the head of the FAMULUS master file being read. Some examples:

```
/ID/LAB NOTES--PATHOLOGY PROJECT
/ID/PSW F&R EXPERIMENT STATION LIBRARY CATALOG
/ID/UNIV. OF GEORGIA BULLETIN
```

3.3.2 /FIELDS/ Card (required)

This card specifies the order into which the fields are to be rearranged. Immediately following the /FIELDS/ card should come the label of the fields to be sorted on, arranged in the desired order and enclosed in parentheses. Those fields not specified on the /FIELDS/ card will remain in the order they are on the master file being read. The /FIELDS/ card may include as many as ten or as few as one field

label. Punching of the control card begins in the first column and continues as far as necessary with each label separated by commas but with NO spaces. No more than one card is ever necessary for a /FIELDS/ statement.

```
/FIELDS/(AUTH,TITL,DATE)
/FIELDS/(LAB,NOTE,DATE)
/FIELDS/(JOB,GRADE,NAME,SSAN)
```

3.3.3 /SELECT/ Card (optional)

This card is used to break a large file into sections for more efficient sorting. If your file exceeds 3,000 records, it is more efficient to sort the first half in one run, sort the second half, and then merge the two output tapes. For collections exceeding 8- o. 9,000 records, it would be more efficient to break the file into 3 parts. Control card punching begins in Column 1 and following /SELECT/ is the sequence to be sorted enclosed in parentheses. For example:

```
/SELECT/(1-1500)
/SELECT/(1501-3200)
```

To sort the entire file, omit /SELECT/.

3.4 THE MERGE PROGRAM

This program merges two master files to create a third. The master files merged must have the same number of fields, but the field labels do not have to be the same. One common use of MERGE is to combine files kept by different researchers. For instance, the various members of a working group may maintain separate literature files, but on occasion they may wish to make a master listing of all these files.

The first master file is merged with the second and the the output is writte to a third. The field labels of the master file created will be the same as on the first master file, unless a /NEW ID/ card is used.

Any record which is not in correct alphabetical order will cause the MERGE program to halt processing, reject the remainder of the job, and print an error message. Therefore, only master files which have been sorted into the same order, by using the SORT program, should be merged.

The MERGE Program prints the /ID/ of the master file, the /ID/ of the additions tape, and the /ID/ of the output tape along with the field labels for each. This will be followed by the number of

records on the newly updated tape and the familiar three large "OK's". If the /PRINT/ control is used, the program will produce a listing of the merged master file.

The program control cards acceptable to MERGE are (parameter "c" represents "column number"):

```
/ID/1st file name      (required)
/ID/2nd file name      (required)
/NEW ID/3rd file name (optional)
/PRINT/(c)             (optional)
```

3.4.1 /ID/ Cards (required)

This program requires two /ID/ cards be used. The first must match the /ID/ name on the first master file; the second must match the second. Some examples:

```
/ID/UNIV. OF GEORGIA BULLETIN
/ID/CROSS REFERENCES
```

3.4.2 /NEW ID/ Card (optional)

This card changes the name written at the head of the new (merged) master file created. If that card is omitted, the name written at the head of the new file is the same as on the /ID/ card for the first master file--the same as the first /ID/ card in the job.

The /NEW ID/ card is in the same format as the /ID/ CARD. When a /NEW ID/ card is used, the /ID/ cards used with programs reading the file created must bear the name given on the /NEW ID/ card. If two files are merged--/ID/JONES NOTES and /ID/SMITH NOTES--and this card is also used--/NEW ID/MERGED NOTES--then /ID/ MERGED NOTES must be used with the GALLEY program used to print off a listing of the newly created master file.

3.4.3 /PRINT/ Card (optional)

If this statement is included exactly as shown above the entire collection will be printed 128 columns wide. If a different page width is desired, then the selected width (in columns) should be placed after the /PRINT/ enclosed in parentheses, e.g., /PRINT/(65) will produce a printout 65 columns wide, approximately notebook paper width.

3.5 THE OSSIFY PROGRAM

The OSSIFY program writes a copy of a FAMULUS master file in input format--80 column lines with each field beginning on a new line and continuations beginning in Column 6. That is the opposite of the action of the EDIT program which takes files in input format and writes their contents into a master file. When FAMULUS was developed, OSSIFY was provided to allow users to create a "backup deck" of punched cards from a FAMULUS tape. If you keep a copy of your input file on a tape or disk--as I recommend--you will not need OSSIFY to create one. However, if you accidentally wipe out your input file, you can recreate it using OSSIFY.

OSSIFY accepts only two control cards (parameter "n" stands for "record number):

```
/ID/file name          (required)
/SELECT/(n1,n2,...)    (optional)
```

3.5.1 /ID/ Card (required)

This card must match the name of the file being read. Some examples:

```
/ID/LAB NOTES--PATHOLOGY PROJECT
/ID/PSW F&R EXPERIMENT STATION LIBRARY CATALOG
/ID/UNIV. OF GEORGIA  BULLETIN
```

3.5.2 /SELECT/ (Optional)

This card is used to write part of a master file. The numbers of the records to be written follow the /SELECT/ on the card, enclosed in parentheses. For example:

```
/SELECT/(1-2500)
/SELECT/(2501-6000)
```

3.6 THE GALLEY PROGRAM

GALLEY produces listings of FAMULUS master files. It prints the fields in each record in sequence without starting a new line for each field. If the /PRINT BY FIELDS/ option is used, GALLEY prints a listing showing the field labels used with each label left justified on a new line. That printout is like the one produced by the EDIT program.

GALLEY recognizes these program control cards (parameter "c" stands for "column number" and "n" for "record number"):

/ID/file name	(required)
/PRINT BY FIELDS/	(optional)
/WIDTH/(c)	(optional)
/SELECT/(n1,n2,...)	(optional)

3.6.1 /ID/ Card (required)

This card must be present and must match the name of the file being read. The /ID/ cards look like this:

```
/ID/FAMULUS BIBLIOGRAPHY
/ID/PSW F&R EXPERIMENT STATION LIBRARY CATALOG
/ID/UNIV. OF GEORGIA BULLETIN
```

3.6.2 /PRINT BY FIELDS/ Card (optional)

This card causes the GALLEY to list the master file in the same format as in the listing produced by the EDIT program; that is, the field labels are shown and each new field begins on a new line.

3.6.3 /SELECT/ Card (optional)

This card allows the user to designate which records are to be printed. Any portions of the collection may be selected. For example:

```
/SELECT/(1-100,7500-1000,1200-1500)
```

3.6.4 /WIDTH/ Card (optional)

This card allows the user to specify the printed page width. It may range from 20 characters (columns) to 128. There are about 10 characters per inch on typical printout paper. The card /WIDTH/(65) will produce a printout 65 columns wide, approximately notebook width.

3.7 THE VOCAB PROGRAM

The VOCAB program will scan any field or set of fields in the records in your file and compile an alphabetized list of terms occuring in the fields. These lists can be used for editing purposes, for constructing search formulas, or for thesaurus building. A list of the words in the title field and of terms in the descriptor field is an effective subject outline of your collection. Using these lists, you can construct search formulas as explicit as you feel necessary. It is advisable to have a separate list of words from each of the different fields scanned by VOCAB. VOCAB can compile one combined list from two or more fields, but you will not then know from which field the word is taken. That would make constructing effective search formulas more difficult (See Chapter 4 for information about effective searching).

To exclude "meaningless" words, such as articles and prepositions, VOCAB uses a stop-word feature. The stoplist consists of a file containing the terms to be excluded from the printout. The stop list must be used each time VOCAB is run. Standard stop-word lists are available. You may also want to add words to such a list.

VOCAB is not a self-starting program. Before it can compile a vocabulary list, it must have a token list of at least one word. The best procedure is to make a token list consisting of the first few meaningful words found in any record in the file.

Both the stop-word list and the token vocabulary list are part of the program control card input. However, these lists can be changed after each run. The VOCAB program produces printed and punched output. The printed output consists of the alphabetical lists of words and a summary of word-length frequencies. The punched output consists of a newly punched stop-list, and a card deck equivalent of the vocabulary list. This deck can be used as input to the next VOCAB run or, or if you prefer, you may make additions to the existing deck or develop an entirely new deck.

The control card input for VOCAB consists of the stop-word and vocabulary deck, an /ID/ card, /FIELDS/ card, and a /SELECT/ card. The /SELECT/ card allows the user to scan only part of the collection. This option is helpful when very large or heavily-indexed collections are being scanned. The dictionary limits on vocabulary size are either 4,500 entries or 45,000 characters, whichever is exceeded first. If

more than one field is being scanned in a single job, this limit can be quickly reached with only 1,500-2,000 citations. This is another reason for scanning one field per job. For very large files, scanning only one field per job can exceed the program's limits. When this happens, the /SELECT/ option is available. If VOCAB's dictionary limits are exceeded, you can locate the point where capability was reached. This is indicated by a message immediately preceding the statistics section (see the note at the end of the VOCAB section) and following the actual word lists. The message indicates the last citation examined. By running another VOCAB job using the select option, vocabulary for the remaining citations can be obtained.

The control cards used by VOCAB are (parameter "f" represents "field label"; "n" represents "record number"; and "t" represents "term"):

```
/ID/ file name          (required)
/FIELDS/(f1,f2,...f10) (optional)
/SELECT/(n1,n2,...)     (optional)
/VOCABULARY/t1,t2,...   (required)
```

3.7.1 /ID/ Card (required)

The /ID/ card must identify the file to be scanned and therefore must match the /ID/ record written at the head of the master file. Some examples:

```
/ID/FAMULUS BIBLIOGRAPHY
/ID/PSW F&R  EXPERIMENT STATION LIBRARY CATALOG
/ID/UNIV. OF GEORGIA BULLETIN
```

3.7.2 /FIELDS/ Card (optional)

The /FIELDS/ card indicates the fields to be scanned. If it is omitted, only the descriptor field is scanned. Some examples of /FIELDS/ cards:

```
/FIELDS/(TITL)
/FIELDS/(TITL,NOTE,DESC)
/FIELDS/(SUBJ)
```

3.7.3 /SELECT/ Card (optional)

The /SELECT/ card specifies the portion of the file to be scanned. The numbers of the records to be scanned, in parentheses, are placed on the /SELECT/ card in the same manner as when using the /SELECT/ card with the SORT program. Some examples:

```
/SELECT/(1-1500)
/SELECT/(1501-3200)
```

If the /SELECT/ card is omitted, the entire file is scanned.

3.7.4 /VOCABULARY/ Card (required)

The vocabulary and stop-word lists are punched on a /VOCABULARY/ card. The vocabulary words and the stop-words may be intermingled. The stop-words are distinguished by an asterisk (*) on the end of each. All entries are separated by commas. Blanks can be inserted anywhere except within a word. Punching continues from 80 directly to Column 1 on the next cards. You may omit the comma following a stop-word because the asterisk suffices as a separator.

Stop-words do not occur in a descriptor field because the program looks at the phrases found in that field, not the individual words. Examples of /VOCABULARY/ cards:

```
/VOCABULARY/A*AN*THE*OR*AND*GEOLOGY,SEDIMENTATION
/VOCABULARY/SCHOOL,DAPARTMENT
```

3.7.5 Statistics Produced

Explanation of statistics for output dictionary section of the VOCAB printout.

Two kinds of statistics are written--figures for the stop-words and figures for the actual vocabulary terms. The stop-word-length statistics are self-explanatory. For vocabulary terms, the table of word-length distributions gives the number of terms of each length occuring in the fields scanned. The table has four rows and ten columns. The first row gives the values of terms having one to ten characters; The second, for 11 to 20 characters; the third, for 21-30 characters; and the fourth, for 31 to 40 characters. The average and maximum lengths of vocabulary entries are given along with total number of entries and the total number of entries and total number of characters for the entries.

3.8 THE INDEX PROGRAM

This program compiles an alphabetical index by scanning the descriptor field and printing each descriptor and a list of the record number where it occurs. INDEX operates only on the descriptor field. Descriptor words or phrases longer than 40 characters are truncated during the printing of the index. The capacity of the program is limited to processing 2,000 different descriptors referenced up to 2,500 times. INDEX accepts the /SELECT/ card which allows the user to determine which portion of the collection will be operated upon. INDEX also has a punch option, which allows the user to have the index written to cards or a computer file. The files containing the various portions of the index can then be merged to provide a master index to the complete file.

INDEX program control cards are following (the parameter "n" represents "number"):

```
/ID/file name          (required)
/SELECT/(n1,n2,...)    (optional)
/PUNCH/                (optional)
```

3.8.1 /ID/ Card (required)

This card must match the file name as it was written in the master file. Some examples of /ID/ cards:

```
/ID/LAB NOTES--PATHOLOGY PROJECT
/ID/PSW F&R EXPERIMENT STATION LIBRARY CATALOG
```

3.8.2 /SELECT/ Card (optional)

This card specifies the portion of the file to be indexed. The record numbers, for example 1-500, or 501-1000, should follow /SELECT/ and be enclosed in parentheses. For example:

```
/SELECT/(1-500)
/SELECT/(501-1000)
```

3.8.3 /PUNCH/ Card (optional)

This statement is used by itself, without sequence numbers, to write the index as a deck of punched cards (or into a file).

3.9 THE SEARCH PROGRAM

SEARCH is the information retrieval program of FAMULUS. The program scans the master file and retrieves (prints out) those records whose field contents meet the criteria of the search question. The "question" is a "search formula", which is constructed according to Boolean principles (see Chapter 4, EFFICIENT USE OF THE SEARCH PROGRAM). If the /WRITE TAPE/ control card is used, the SEARCH program will write the results of the search into a new FAMULUS master file.

The SEARCH program will operate on any field in a record. In bibliographic records, it will most likely be called upon to examine descriptor, title, author, and date fields.

A satisfactory search result depends on factors which are almost wholly in your control. You must have a clear concept of what you wish to retrieve from your file. Equally important is your understanding of how the search logic works and your skill in constructing search formulas. Also, effective searching requires that the records be of high quality as explained in Chapter 2, Section 2.3 Quality Control. Inept original indexing cannot be compensated for by the logic of the SEARCH program.

Any field or combination of fields may be searched for any or all of the specified terms. Because of the flexibility of the program, there are usually several different search formulas which will retrieve the same set of records.

SEARCH provides the option of writing the retrieval set of records into a file for future use. The GALLEY program can be used to print a listing of the file.

Program control cards acceptable to SEARCH follow (parameter "f" stands for "field label"):

```
/ID/file name              (required)
/FIELDS/  (f1,f2,...f10)   (optional)
/WRITE TAPE/               (optional)
/SEARCH/a search formula   (required)
```

3.9.1 /ID/ Card (required)

This card must match the file name written on the master file to be searched. Some examples of /ID/ cards:

```
/ID/FAMULUS BIBLIOGRAPHY
/ID/PSW F&R EXPERIMENT STATION LIBRARY CATALOG
/ID/UNIV. OF GEORGIA BULLETIN
```

3.9.2 /FIELDS/ Card (optional)

This card lists the labels of the fields to be searched for retrieval terms. The labels should follow the /FIELDS/ statement and be enclosed in parentheses. If the entire record is to be searched (a very inefficient approach), no /FIELDS/ card is necessary. Examples:

```
/FIELDS/(TITL,DESC)
/FIELDS/(ONAM,TITL,SNAM,NOTE)
```

3.9.3 /WRITE TAPE/ Card (optional)

This statement causes the records selected to be written to a newly created FAMULUS master file. Of course, the new master file contains a subset of the records in the original master file. The new master file can be used like the master files created by EDIT, SORT, or MERGE. If you want to edit the contents, you can use OSSIFY to dump the contents of the new master file into an input file.

When /WRITE TAPE/ is used, the master file created must be accounted for in the "DD" descriptions included in the job control cards (See Appendix D, Section D.1.3.2, DD Statements). When /WRITE TAPE/ is used, only one /SEARCH/ request can be made.

```
/WRITE TAPE/
```

3.9.4 /SEARCH/ Card (required)

More than one search may be made by using more than one /SEARCH/ card. The search formula which appears on the /SEARCH/ card tells the computer which terms to search for. In search formulas, the terms used are joined by the logical operators.

They are:

Meaning	Symbol	IBM Code
and	&	2-8
or	|	11-8-7
not	¬	12-8-7

Those symbols are not available on all terminal keyboards. However, sometimes one symbol on a particular terminal is equivalent to another. The technical manual on your terminal may help you find an equivalent symbol. Also, the communications specialists at the Computer Center may know what symbols are equivalent. If you cannot find the symbols on your terminal and you cannot find another terminal with them on it, you can dump your SEARCH job onto IBM cards and punch the /SEARCH/ card using an IBM 29 Card Punch machine. The keyboard for that key-punch machine contains the special symbols needed for expressing SEARCH formulas.

Use of the AND (&) operator implies that the user wants to retrieve only records in which all of the terms joined by AND are present. If one of the terms is missing, the record will not be retrieved. The OR (|) operator will retrieve records in which any of the specified terms are present. The NOT (¬) operator specifies that records in which the term preceded by NOT is present will not be retrieved.[13]

Truncated versions of terms can be used to retrieve records having those terms in various forms. For instance, if a VOCAB program listing shows that the term PROGRAMMING occurs with both one M and two M's, then use of a truncated form, namely PROGRAM would retrieve both forms. Othewise, both PROGRAMMING and PROGRAMING would have to be included in the search formula joined by the OR operator.

The terms are searched for in the fields specified by the /FIELDS/ card. Sometimes there is a need to include a term from another field, such as an author or date. In that case, append the field label to the term (enclosed in parentheses). For example:

SMITH(AUTH) or 1948(DATE)

---

[13] The use of the logical operators in search formulas is described and illustrated at length in Chapter 4, EFFECTIVE USE OF THE SEARCH PROGRAM.

The truncation feature can be used to select a range of dates. For example, (ECOLOGY | GEOLOGY | HYDROLOGY) & 197(DATE) retrieves articles on the topics shown but only those with dates dates 1970 through 1979.

A search formula may have no more than 50 different descriptors (or, 60 including duplicates) with a total of 150 descriptors and operators. There is no limit on the depth to which phrases can be nested--set off by parentheses.

Chapter 4

EFFECTIVE USE OF THE SEARCH PROGRAM

You do not need formal knowledge of symbolic logic or Boolean algebra to make good use of the SEARCH program, but you must understand the basic search operations and the results they produce. Also, you must understand how to translate your questions into FAMUS search formulas.

4.1 CONJUNCTION

The simplest thing you can ask the SEARCH program to do is to give you a printout of all the records in your file which have a given term in the fields searched, for example, PINUS LAMBERTIANA in the descriptor field. This would result in a printout of the subset of your file containing all your information on Sugar pine. You can narrow the search further. Suppose you were looking for material on embryo dormancy in Sugar pine. You could then instruct the SEARCH program to printout those records containing the keywords PINUS LAMBERTIANA and EMBRYO DORMANCY. That will be a smaller subset. The search may be made as specific as needed. The example below shows how natural language questions can be translated into search terms.

Question	Search Terms
1. "Print out everything on P. lambertiana."	PINUS LAMBERTIANA
2. "Print out everthing on embryo dormancy in Sugar pine."	PINUS LAMBERTIANA & EMBRYO DORMANCY
3. "Print out everything on meristemic regeneration from dormant Sugar pine embryos."	PINUS LAMBERTIANA & EMBRYO DORMANCY & MERISTEMIC REGENERATION
4. "Print out everything on in vitro regeneration of root meristems of dormant Sugar pine embryos."	PINUS LAMBERTIANA & EMBRYO DORMANCY & MERISTEM REGENERATION & IN VITRO

Note that each term added (or &'ed) to the search formula will reduce the size of the subset retrieved.

The search formulas are commutative--the terms may be listed in any order, separated by logical signs, for example:

EMBRYO DORMANCY & IN VITRO & PINUS LAMBERTIANA & MERISTEM REGENERATION

4.2 DISJUNCTION

This operation allows you to get a printout of those records which contain either one of two terms or both in the designated fields. For example, you might wish to see those records containing either EROSION or SLIDES or both. Disjunction is used with conjunction to develop very specific search formulas.

Suppose that you had a FAMULUS master file on national forests and national parks and that you wanted to see all records on soil compaction resulting from recreational use in either. If you have used NATIONAL FORESTS and NATIONAL PARKS as descriptors, you will want to see all records which refer to either. To ask for NATIONAL FORESTS and NATIONAL PARKS would retrieve only those records which treat both. Probably most of your records treat only one or the other. Your search strategy must be to ask: "Print out everything on national forests or

national parks and recreation use and soil compaction." This search formula would do that:

(NATIONAL FORESTS | NATIONAL PARKS) & RECREATION & SOIL COMPACTION

Note that the two terms joined by the OR (|) sign (the disjunctive operator) are enclosed in parentheses.

4.3 CONSTRUCTING THE SEARCH FORMULA

The keywords or terms used in the search formula may be drawn from a vocabulary list. That list may have been generated by the VOCAB program from terms you have put into the descriptor field. In that case, you are working with a "controlled vocabulary" -- you are using terms which you put into the records specifically as indexing terms. You may also have VOCAB generate a list of terms found in a field besides the descriptor field, like the title field. That is an "uncontrolled vocabulary" or "natural language vocabulary" because the terms appear in the records "naturally" and were not supplied to aid in searching. Of course, you should use the descriptor terms to search the descriptor field and terms from another field to search that field.

If you want to search both descriptor and another field at the same time, you should construct the search formula as if it were for a non-descriptor field. That is because you must allow for all grammatical variations of natural language when you are working with the uncontrolled vocabulary in the non-descriptor field. In many titles, for example, common rather than botanical names appear. A phrase like "dormancy of embryos" might appear instead of EMBRYO DORMANCY. Working with natural language vocabularies requires use of truncation and of the disjunctive operator. The examples of conjunctive searches shown above used keywords from a controlled descriptor vocabulary. Question four, with truncated terms, could use this search formula:

PINUS LAMBERT & EMBRYO DORMAN & MERISTERM REGEN & IN VITRO

If you are searching a title field for the same information, or in conjunction with a descriptor field, you will have to employ the disjunctive operator in order to catch deviant forms in the titles:

(PINUS LAMBERT | SUGAR PINE) & (EMBRYO & DORMAN) & (MERISTEM & REGEN) & IN VITRO

That search formula asks: "Print out everything on PINUS LAMBERTIANA or SUGAR PINE and EMBRYO[S/NIC] and DORMAN[T/CY] and MERISTEM[IC/S] and REGEN[ERATE/ERATION/ERATING] and IN VITRO." That will catch deviant forms in titles. It may increase the "noise" in your printout by pulling in some terms which are not germane to your question. The

number of unwanted records which that formula would retrieve would probably be small.

4.4 FORMULA STRUCTURES

The basic structures illustrate possible search formula structures. As in algebraic formulations, increasingly complex structures can be elaborated. It is seldom likely that you will have to develop questions more complex than the ones shown.

Formula	Result
A	Prints all records containing A (either by A alone or with other terms).
A &¬ B	Prints all Records containing A and any other terms except B.
A & (¬B & ¬C)	Prints all records containing A and any other terms except B and C.
A & B	Prints all records containing both A and B. This structure may be extended as far as necessary, e.g., A & B & C & D, etc,
A | B	Prints all records containing A or B or by both. This structure can be extended as far as necessary, e.g., A | B | C | D, etc.
A & (B | C) & D	Prints all records containing both A and D, and B or C.
A & B & ¬(C | D)	Prints all records containing both A and B and not C or D.

4.5 EXAMPLES OF SEARCHES

A & (B | C) & D

/FIELDS/(TITLE,KEY) /SEARCH/SOIL STABILITY & (EROSION | SLIDES) &
198(DATE)

This search formula will cause the computer to scan the title and keyword files and retrieve any records containing the phrase SOIL STABILITY and either EROSION or SLIDES. Only records dated 1980 or later will be retrieved.

(A | B | C | D) & E & F

/FIELDS/(DESC)
/SEARCH/(PANCHROMATIC | INFRARED | THERMAL | RADAR) &
AERIAL & 70MM

This formula will cause the computer to scan the descriptor field and print any records including AERIAL and 70MM and any of the terms in parentheses--PANCHROMATIC, INFRARED, THERMAL, or RADAR.

((A & B) | (C & D) | E) & (F | G | H |I)

/FIELDS/(TITL)
/SEARCH/((CLEAR & CUTTING) | (SLASH & DISPOSAL) |
SCARIFICATION) & (LODGEPOLE | PONDEROSA | DOUGLAS-FIR
| LONGLEAF)

This formula will retrieve records with either CLEAR CUTTING, SLASH DISPOSAL, or SCARIFICATION in the title combined with either LODGEPOLE, PONDEROSA, DOUGLAS-FIR, or LONGLEAF. By using SLASH | DISPOSAL, you avoid missing items which have "disposal of slash" in their titles.

(A | B) & C & D

/FIELDS/(TITL,DESC)
/SEARCH/(SPECTROPHOTOMETRY | RADIOMETRY) & CELL STRUCTURE &
JONES(LOC)

This search formula will retrieve items including either SPECTROPHOTOMETRY or RADIOMETRY and CELL STRUCTURE which have the name "Jones" in the location field. In other words, the output will be a list of items in "Jones" collection on the subject of SPECTROPHOTOMETRY or RADIOMETRY applied to the study of cell structure.

Appendix A

RULES FOR KEYPUNCHING

A.1 PROGRAM CONTROL CARDS

All control cards begin with a slash in Column 1. The control statement follows the slash and is closed by a second slash, e.g., /ID/ or /FIELDS/. This rule is invariable. Most control statements are followed by a text which contains the particulars to be handled by the program.

The text following many statements is enclosed by parentheses:

```
/FIELDS/(AUTH,TITL,PUB,DATE)
/DELETE/(3,15-19,46-68,101)
/SELECT/(1-500,575-690)
```

The specific format for the text following a given statement is shown in Chapter 3 at the point the statement is explained.

Some statements are not followed by text, e.g., /ORIGINAL/,/CITATIONS/.

All text, with or without parentheses, is punched following the second slash of the control statement. When it is necessary to continue the text onto a second card, it begins in Column 1 of the second card.

A.2 RECORDS

Records may be up to 4,000 characters long, excluding field labels of which there may be ten or fewer. Since the number of characters placed on a card will vary, there is no fixed number of lines which may appear in one record.

Each record in the input file must be followed by a blank line. The blank line tells the EDIT program that a new record is beginning. Failure to leave a blank line is a very common error. It results in two records being rejected.

Field labels are punched in Columns 1-4 of the first card in the field. Continuing cards in a field do not carry labels. Actual text of the record begins in Column 6 and continues through Column 80. If it is necessary to continue onto another card, punching begins in either Column 6 or Column 7. If the last word of the preceding card ended in Column 80, then punching begins in Column 7. If the word was broken in the middle, or if Column 80 was a blank separating two words, begin in Column 6. Do not use a hyphen in Column 80 to break a word if it is not complete; simply punch up through Column 80 and place the remainder of the word on the next card beginning in Column 6. For example, if a title ended with ". . .THE UNIVERSITY OF GEORGIA" and the "Y" were in Column 80, Column 6 on the next line would be blank:

```
TITL PROGRAM FOR THE EDUCATION OF . . .
      OF GEORGIA.
```

If the "T" in "UNIVERSITY" were in Column 80, the "Y" would be in Column 6 on the next line:

```
TITL PROGRAM FOR THE EDUCATION OF . . .
     Y OF GEORGIA.
```

FAMULUS will put split words together at the time of printing.

The FAMULUS SORT program does not evaluate numbers as number--only as characters. For that reason, SORT will put number "9" ahead of "88 because "9" is larger than--falls ahead of--the first "8". SORT would put " 9" (or "09") and "88" in the proper order. If a field is set aside for numbers, the user should use leading blanks to make the numbers sort properly. For example, the serial number in several records might look like this in an input file:

```
SERI      1
SERI     22
SERI    333
SERI   4444
SERI  55555
```

Following are some examples of different types of FAMULUS records.

Example 1:

```
AUTH COMPERE, H.
TITL THE RED SCALE AND ITS INSECT ENEMIES.
DATE 1961.
PUB  HILGARDIA, 31(7),173-278.
KEY  BIOCONTROL, MASS CULTURE, COLONIZATION, HYPERPARASITES
LOC  PROJECT FILE A
```

Example 2:

```
NAME SMITH, MARY
TITL CLERK TYPIST
SSAN 546-83-8289
GS   GS-4(CAREER CONDITIONAL)
DOB  7/21/60
```

Example 3:

```
NOTE LAB NOTE 143
LAB  LAB 5,BERKELEY
DATE 06/13/79
TEST EFFECTS OF SMOG ON PONDEROSA PINE--THIRD STAGE TESTING
RMRK RESULTS--DECREASE IN NEEDLE LENGTH AND WATER CONTENT
```

Example 4:

```
F1YR 1913-1914.
VLNU VOL. 14 NO.  4.
TITL NEGROES IN ATHENS, GEORGIA, BY T.J. WOOFTER, JR.
SNAM PHELPS-STOKES FELLOWSHIP STUDIES, NO.  1.
F9SN SERIAL NO.  217.
NOTE NOTE: STUDY SUPERVISED BY PRESTON BROOKS.
```

Appendix B

HOW TO INTERPRET ERROR MESSAGES

The EDIT Program

The two most frequently occuring error messages are:

/ID/ DOES NOT MATCH ID ON INPUT TAPE

and

FIELDS ARE OUT OF ORDER SKIPPING THIS CITATION.

The first is self-explanatory. The second message occurs if the fields are actually out of order or if the blank line between citations is omitted. The entire group of cards which was skipped is set off with asterisks.

EDIT also prints the following messages:

ERROR IN ABOVE STATEMENT -- which refers to the control card immediately preceding the error message.

THIS STATEMENT DOES NOT BELONG IN EDIT -- an illegal program control card has been included.

The other EDIT program messages are self-explanatory.

The SORT Program

Again, the message /ID/ DOES NOT MATCH ID ON INPUT TAPE is the most common. The other messages refer primarily to the /FIELDS/ statement and indicate if it has been omitted, if it declares fields which do not match those on the tape, or if it is incorrectly punched.

The MERGE Program

MERGE checks both /ID/'s to see if they match the input file name and prints an error message if they do not.

It also will print an error message and halt if it encounters records which are not in alphabetical order. The program will identify the record which is out of order and will terminate processing. The file which is out of order must then be sorted to get it into correct order. Then MERGE can be run again.

The GALLEY Program

GALLEY recognizes errors on the /ID/ card or any incorrectly included program control cards.

The INDEX Program

INDEX checks for matching /ID/ statement and file name. The only other messages it prints concern whether the /DESCRIPTOR FIELD/ was correctly defined and whether there are included any extraneous program control cards.

The VOCAB Program

Again, the /ID/ message will be written if the /ID/ is not identical to the one on the master file. If a /FIELDS/ statement is not found or is incorrectly punched, a message will result. The program prints a message when more terms are found than its dictionary can hold. If this occurs the last citation examined by this run of the program will be listed in the statistics section following the vocabulary lists. Another run can then be set up to process the remaining records.

The SEARCH Program

After checking the /ID/ statement, SEARCH scans the search formula to insure that the logical operators are correctly punched and that parentheses are balanced. If they are not, it will print an error message and terminate the job. It also checks to see that the search formula is valid.

The OSSIFY Program

OSSIFY checks the /ID/ and then scans any number given in the /SELECT/ statement. If there is a punching error, a message will be printed and execution will halt.

Appendix C

JOB CONTROL LANGUAGE

FAMULUS runs on the IBM main-frame computer and Job Control Language (JCL) controls that computer. The JCL tells the computer:

1. That the user is authorized to submit jobs.
2. What programs to execute.
3. What files to read and write.
4. What equipment--card readers, disk drives, tape drives, etc--to use.
5. How to do the job.

An error in the JCL will probably cause the job to fail. The computer is not forgiving. If you put a letter O where a 0 is required or the letter l in place of 1, your job will surely fail. The purpose of this appendix is to explain the particular JCL needed to run the FAMULUS programs--not JCL in general.[14]

A person not familiar with the IBM system should read Chapter 5, IBM JOB CONTROL LANGUAGE (JCL), in the FACILITY ACCESS AND SERVICES GUIDE[15] before attempting to run the FAMULUS programs.

The Technical Support Office of the computer center, 542-3106, makes consultants available to help people in the University community make good use of the computer facilities. In addition to consulting, Technical Support periodically gives seminars on JCL and other topics free of charge. You can enroll by calling 542-3106. A new computer user at the University of Georgia should be sure to ask about KLEAN in Technical Support. KLEAN is a program which can automatically wipe out data sets. It is described in DATASET PROTECTION ON THE IBM (Available from the computer center as Memo "108").

---

[14] A good introduction to JCL is: Brown, Gary D. System/370 Job Control Language. New York: John Wiley & Sons, Inc.; 1977.

[15] That GUIDE and other publications which may be useful to the FAMULUS user are listed in the REFERENCES at the end of this manual.

C.1 THE FAMULUS JOB

A job consists of:

1. A job card.
2. JES3 control statements, if any.
3. One or more steps.

Those must all be in the proper JCL syntax (see Section 5.2 in the FACILITY ACCESS AND SERVICES GUIDE).

C.1.1 The JOB Card

Each job has a job card. The Job card tells the computer that the user is authorized to submit the job and it can be used to tell the computer how and when to submit the job. Also the job name on the job card appears on the "banner" page at the first of the printout and is used to file the printout. For example, a job with job name DOGS1V1 would be filed in the bin marked "D" in the computer center. The user would look in that bin when picking up the job. See Section 5.4 of the FACILITY ACCESS AND SERVICES GUIDE for more information.

C.1.2 The JES3 Statements

JES3 statements are used to communicate with the system operators. FAMULUS users might use them to instruct an operator to mount a particular tape, to route a printout to a particular printer, or to execute the job at night when it would cost less to run. See WHAT YOU SHOULD KNOW ABOUT JES3 CARDS (listed in the REFERENCES) for more informaion.

C.1.3 The Steps

The step consists of a step card containing an EXEC statement and a number of DD cards telling the computer what files or devices to use, and, optionally, data for the computer to process. A job may have several steps. For example, one step might create a file, another could print a lis ing of the file, another could sort the file and create a new file, the next step could print a listing of that file, etc.

C.1.3.1 The Step Card

The step card tells the computer what program to execute. The FAMULUS programs are:

EDIT
SORT
MERGE
OSSIFY
GALLEY
VOCAB
INDEX
SEARCH

The step card is in the form:

```
//stepname EXEC PGM=program,REGION=256K
```

And that card must be followed by a STEPLIB card:

```
//STEPLIB DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
```

The parts in small letters in the examples above are provided by the users, and the parts in capital letters must appear as shown.

The stepname distinguishes one step from another if a job has more than one. The simplest thing to do is to number them sequentially STEP1, STEP2, STEP3, STEP4, etc.

The program name is supplied by the user, depending on which FAMULUS program is to be executed.

The STEPLIB card which must appear as shown tells the computer where to find the FAMULUS program to be executed. Here is a sample showing the step card and the STEPLIB card needed to execute the GALLEY program:

```
//STEP1 EXEC PGM=GALLEY,REGION=256K
//STEPLIB DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
```

C.1.3.2 DD Statements and Unit Numbers

DD, data description, statements are the most complex part of JCL. They tell the computer what peripheral equipment to use--to write a file to a tape on a tape drive, to read data from a disk file, to write a report to a printer, to read data from a card reader, etc. I cannot give more than a brief explanation of them and some examples here. Users should consult Section 5.6 in the FACILITY ACCESS AND SERVICES GUIDE and the Brown, System/370 Job Control Language, for more detailed information. Also the consultants available through the computer center, 542-3106,

can advise you about setting up and using files and peripheral equipment.

DD statements consist of a label, DD, and parameters. Like:

//label DD parameter1,parameter2,etc

In the STEPLIB statement mentioned above, STEPLIB is the label and the parameters tell the computer that the program to be executed is in a data set named USERSRV.FAMULUS.LOAD and that the dataset may be read but not altered.

The FAMULUS programs were written in the FORTRAN programming language and the DD statements in FAMULUS jobs, except for the STEPLIB statement, use standard FORTRAN "unit numbers" in their labels.[16] The label tells the computer where to find the file or device that is to be used by the program. Standard FORTRAN labels look like this:

FTnnF001

The number represented by "nn" in the example is the "unit number." In the early days of computing, the unit number told the computer what external unit to use. Often there was only one printer, for example. Now the unit number may refer to a particular device or to a file which can be accessed by the system.

The FAMULUS programs follow a FORTRAN convention of using Unit Number 05 for input from the job stream--the job or file submitted by the user--and Unit Number 06 for output to a printer. FAMULUS also uses a number of other unit numbers. For example, Unit Number 02 is used for the file into which the EDIT program writes the master file it creates. The use of unit numbers is illustrated in this example of a one-step job:

---

[16] In that respect FAMULUS is very similar to the IBM version of the Statistical Package for the Social Sciences.

```
//DOGS1  JOB USER=UGAUSER,PASSWORD=BULLDOG,TIME=1
//*MAIN ORG=LOCAL,LINES=19
//*  EDIT JOB 1
//STEP1   EXEC PGM=EDIT,REGION=256K
//STEPLIB   DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT02F001  DD DSN=UGU1.MASTER1.DATA,DISP=OLD
//FT06F001  DD SYSOUT=A
//FT05F001 DD *
/ID/FAMULUS BIBLIOGRAPHY
/FIELDS/(AUTH,TITL,IMPR,PART,AVAI,NOTE,XREF,SJUB,F9,F10)
/DESCRIPTOR FIELD/(SUBJ)
/ORIGINAL/
/CITATIONS/

AUTH BATTY, C. DAVID.
XREF SEE: MCCRANK--THE MT. ANGEL ABBEY MANUSCRIPT . . .

AUTH BATTY, C. DAVID; DESTRICKER, J.; POLSON, L.
TITL FAMULETTA: A FAMULUS SAMPLER.
IMPR MONTREAL: MCGILL UNIVERSTIY, GRADUATE SCHOOL OF . . .
SUBJ INFORMATION STORAGE AND RETRIEVAL.

etc.
```

In the example:

--The first DD statement identifies the location of the program--in USERSRV.FAMULUS.LOAD.
--The label FT02F001 in the second DD statement tells the computer to write the master file created by the EDIT program into an old file named UGU1.MASTER1.DATA.
--The label FT06F001 in the third DD statement tells the computer to write messages and listings produced by the program to a line printer.
--The label FT05F001 in the fourth tells the computer that the data to be read follows immediately after that card. In this case, program control cards and FAMULUS records follow.

If the FAMULUS control cards and records to be used were in file UGU1.FAM1.DATA, the fourth DD statement could have looked like this:

```
//FT05F001 DD DSN=UGU1.FAM1.DATA,DISP=SHR
```

C.1.3.3 Unit Numbers Used by FAMULUS Jobs

Jobs to run FAMULUS programs require different sets of DD statements. All require DD statements with Unit Numbers 05 and 06, and all but EDIT require a DD statement with Unit Number 01.

01 The master file. All programs, except EDIT, read a master file.

02 The master file created by EDIT, SORT, and MERGE and by SEARCH if the /TAPE WRITE/ option is used.

03 The second master file read by the MERGE program.

05 The control cards used by all of the programs. The records read follow the control cards in EDIT jobs.

06 The listings produced by all of the programs. Those listings can be printed on green-bar computer paper or on letter-size docuform paper, or it may be routed to the "laser printer" (officially the "IBM 6670 Information Destributor).

07 The output from the OSSIFY program and from the INDEX program (if the /PUNCH/ option is used). If "SYSOUT=B" is used, the output will be punched on cards.

11, 12, 12, 14 The working storage units units used by the SORT program.

Unit Numbers 01, 03, and 05 represent files read by programs.

Unit Numbers 02, 04, and 06 repersent devices--like tape drives--that are written to.

Unit Numbers 11-14 are temporary files used during processing by the SORT program.

C.2 JCL SAMPLES

These examples are based on JCL which has been tested and did work. While your jobs will be similar, they will differ in detail. For example, the master file you create might be smaller or larger than in the examples, you might want your programs to run at night or on the weekends when the computer charges are reduced, or OCIS could change the name of the STEPLIB used. Examples of the following jobs are shown:

EDIT
SORT and GALLEY
MERGE
OSSIFY
GALLEY (using the laser printer)
VOCAB
INDEX
SEARCH
DSDUMP
SORTD

C.2.1 EDIT JOB

This job reads the FAMULUS control cards from the job stream--the job itself--and the records from two files.

```
//DOGS1E JOB USER=UGAUSER,PASSWORD=BULLDOG,
//       NOTIFY=UGAUSER,TIME=2
//*MAIN ORG=NCT19,LINES=12
//*
//*  FAMULUS SYSTEM: EDIT PROGRAM TEST
//*
//STEP    EXEC PGM=EDIT,REGION=256K
//STEPLIB  DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT02F001 DD DSN=UGU1.MASTER1,DISP=OLD
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
/FIELDS/(F1YR,VLNU,ONAM,TITL,SNAM,SNUM,UNAM,UTIL,F9SN,NOTE)
/ORIGINAL/
/CITATIONS/
//       DD DSN=UGU1.FAM1.DATA,DISP=SHR
//       DD DSN=UGU1.FAM2.DATA,DISP=SHR
//
```

C.2.2 SORT and GALLEY JOB

This is a sample of a two step job. Note that the second step reads the master file created by the first step.

```
//DOGS1W JOB USER=UGAUSER,PASSWORD=BULLDOG,TIME=2
//*MAIN ORG=SSS05,LINES=19,USER=UGAUSER
//*
//*  THIS JOB SORTS A FAMULUS MASTER FILE TO
//*  CREATE A NEW MASTER FILE: UGU1.NEWFILE.
//*  THE SECOND STEP PRINTS A LISTING OF THE
//*  NEW FILE.
//*
//STEP1   EXEC PGM=SORT,REGION=256K
//STEPLIB  DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.MASTER1,DISP=SHR
//FT02F001 DD DSN=UGU1.NEWFILE,DISP=(NEW,CATLG),
//     UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=2880,RECFM=FB),
//     SPACE=(TRK,(50,15))
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
/FIELDS/(SNAM,ONAM,TITL,F1YR,VLNU,SNUM,UNAM,UTIL,F9SN,NOTE)
//FT11F001 DD UNIT=SYSDA,SPACE=(2012,(6000,200))
//FT12F001 DD UNIT=SYSDA,SPACE=(2012,(6000,200))
//FT13F001 DD UNIT=SYSDA,SPACE=(2012,(6000,200))
//FT14F001 DD UNIT=SYSDA,SPACE=(2012,(6000,200))
//*
//* * * * * * * * * * * * * * * * * * * * * * * *
//*
//STEP2   EXEC PGM=GALLEY,REGION=256K
//STEPLIB  DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.NEWFILE,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
//
```

C.2.3 MERGE JOB

```
//DOGS1M JOB USER=UGAUSER,PASSWORD=BULLDOG,TIME=2
//*MAIN ORG=SSS05,LINES=19
//*
//*  MERGE PROGRAM JOB
//*  UNITS 1 AND 3 ARE THE FILES BEING MERGED
//*  UNIT 2 IS THE FILE CREATED
//*
//STEP     EXEC PGM=MERGE,REGION=256K
//STEPLIB  DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.MASTER1.DISP=SHR
//FT02F001 DD DSN=UGU1.MASTER3,UNIT=SYSDA,
//            SPACE=(TRK,(5,1)),DISP=(NEW,CATLG),
//            DCB=(LRECL=80,BLKSIZE=2880,RECFM=FB)
//FT03F001 DD DSN=UGU1.MASTER2,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
/ID/XREFS
/PRINT/(65)
//
```

C.2.4 OSSIFY Job

```
//DOGS1O JOB USER=UGAUSER,PASSWORD=BULLDOG,TIME=2
//*MAIN ORG=LOCAL,LINES=10
//STEP1   EXEC PGM=OSSIFY,REGION=256K
//STEPLIB  DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.MASTER1,DISP=SHR
//FT07F001 DD DSN=UGU1.FAM3.DATA,
//         DISP=(NEW,CATLG),
//         UNIT=SYSDA,
//         DCB=(LRECLK=80,BLKSIZE=2880,RECFM=FB),
//         SPACE=(TRK,(50,15))
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
//
```

C.2.5 GALLEY Job--to write to the laser printer

This job prints a listing of the master file on the "laser printer" (officially the IBM 6670 Informatin Distributor). The first step--a "proc" called OCL--sends the laser printer information about the format of the report to be printed. QUANTITY 2 in that step causes two copies to be printed. The second step writes the listing to the laser printer. SYSOUT=Z on the FT06F001 card routes the output to the laser printer.

```
//DOGS1L1 JOB USER=UGAUSER,PASSWORD=BULLDOG,NOTIFY=UGAUSER
//*MAIN ORG=LOCAL,USER=UGAUSER,LINES=19
//*
//*  FAMULUS SYSTEM: GALLEY PROGRAM
//*  JCL FOR PRINTING ON THE LASER PRINTER
//*   +STEP1 GIVES THE LASER PRINTER
//*    INSTRUCTIONS  (CALLED OCL)
//*   +STEP2 PRINTS THE JOB USING THE
//*     GALLEY PROGRAM
//*
//STEP1 EXEC OCL
//SYSUT1 DD *
,PRINT 102
,FIRST LINE 9
,DUPLEX 1
,QUANTITY 2
,END
//*    ********************************
//STEP2  EXEC PGM=GALLEY,REGION=256K
//STEPLIB DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.MASTER1,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
//
```

C.2.6 VOCAB Job

```
//DOGS1V1 JOB USER=UGAUSER,PASSWORD=BULLDOG
//*MAIN ORG=SSS05,LINES=19,USER=UGAUSER
//*
//*  FAMULUS SYSTEM: VOCAB PROGRAM
//*
//STEP1    EXEC PGM=VOCAB,REGION=256K
//STEPLIB DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.MASTER1,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT07F001 DD DUMMY
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
/FIELDS/(ONAM)
/VOCABULARY/SCHOOL,DEPARTMENT
//
```

C.2.7 INDEX Job

```
//DOGS1I JOB USER=UGAUSER,PASSWORD=BULLDOG,
//         NOTIFY=UGAUSER,TIME=2
//*MAIN ORG=NCT19
//*
//*   FAMULUS SYSTEM: INDEX PROGRAM
//*
//STEP1  EXEC PGM=INDEX,REGION=256K
//STEPLIB DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT02F001 DD DSN=UGU1.MASTER1,DISP=OLD
//FT06F001 DD SYSOUT=A
//FT07F001 DD DUMMY
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
//
```

C.2.8 SEARCH Job

Note that more than one search are made in this single step; that is, more than one /SEARCH/ card are included.

```
//DOGS18 JOB USER=UGAUSER,PASSWORD=BULLDOG
//*MAIN ORG=SSS05,LINES=19,USER=UGAUSER
//*
//*  FAMULUS SYSTEM: SEARCH JOB
//*
//STEP1  EXEC PGM=SEARCH,REGION=256K
//STEPLIB DD DSN=USERSRV.FAMULUS.LOAD,DISP=SHR
//FT01F001 DD DSN=UGU1.MASTER1,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT07F001 DD DUMMY
//FT05F001 DD *
/ID/UNIV. OF GEORGIA BULLETIN
/FIELDS/(ONAM,TITL,SNAM,NOTE)
/SEARCH/AGRIC | FARM | RURAL
/SEARCH/FORESTRY
/SEARCH/PHYSICAL ED & WOMEN
//
```

C.2.9 DS DUMP Job--to write a file on a tape

This is not an example of a FAMULUS job. DS DUMP is a handy utility program. It is described in the DATASET DUMP UTILITY USER'S GUIDE listed in the BIBLIOGRAPHY. One common use is to backup--make a copy of--a file. This job reads a disk file and writes it on tape. Note the //*OPERATOR card which asks the operator to mount a certain tape. The SYSUT2 DD card specifies that the output will be written to the tape.

```
//DOGS1D USER=UGAUSER,PASSWORD=BULLDOG
//*MAIN ORG=SSSO5
//*OPERATOR * MOUNT TAPE: VRN=#1234 VSN=DUMMY KEY=ABCD
//STEP EXEC PGM=DSDUMP
//SYSUT1 DD DSN=UGU1.FAM1.DATA,DISP=SHR
//SUSUT2 DD UNIT=TAPE16,LABEL=(3,NL,,IN),DISP=NEW,
//       VOL=SER=DUMMY,
//       DCB=(LRECL=80,BLKSIZE=2880,RECFM=FB)
//SYSIN DD *
COPY='YES',HARDCOPY='NONE',PRINTLAST='YES'
//
```

C.2.10 DSDUMP Job--to create an IBM file

This job creates a file in the IBM system. It was used to transfer a file from the MUSIC system. It could just as well have created a file from card data. Note that the delimiter "##" is used to indicate the end of the input data.

```
//DOGS1DS JOB USER=UGAUSER,PASSWORD=BULLDOG,TIME=1
//*MAIN ORG=SSS05,LINES=19
//*
//*  THIS DUMPS A FILE FROM MUSIC INTO A NEW FILE
//*  WHICH IT CREATES.
//*
//STEP1  EXEC PGM=DSDUMP
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=UGU1.FAM1.DATA,DISP=(NEW,CATLG),
//       UNIT=SYSDA,DCB=(LLRECL=80,BLKSIZE=2880,RECFM=FB),
//       SPACE=(TRK,(50,15))
//SYSIN DD *
COPY='YES',HARDCOPY='NONE',PRINTLAST='YES';
//SYSUT1 DD DATA,DLM=##

F1YR 1935-1936.
VLNU VOL. 36 NO. 01.
. . .
. . .
##
```

C.2.11 SORTD Job

SORTD is not a FAMULUS job. It is another utility program. The program is described in Brown (System/370 Job Control Language) Pages 222-224 and in the FACILITIES AND ACCESS SERVICES GUIDE.

```
//DOGS1SR JOB USER=UGAUSER,PASSWORD=BULLDOG,
//       TIME=2
//*MAIN ORG=LOCAL
//STEP   EXEC SORTD,PARM='SIZE=MAX',REGION=256K
//SORTIN  DD  DSN=UGU1.FAM1.DATA,DISP=SHR
//SORTOUT DD DSN=UGU1.SORTOUT.DATA,DISP=(NEW,CATLG),
//       UNIT=3330,SPACE=(TRK,(3,3),RLSE),
//       DCB=*.SORTIN
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
SORT FIELDS=(5,78,CH,A)
//
```

Appendix D

BIBLIOGRAPHY

D.1 REFERENCES

Section 2.2, DOCUMENTATION SERVICES, in the FACILITY ACCESS AND SERVICES GUIDE listed below describes how to obtain user memos and guides by submitting a jobs to the computer. Most of the publications listed here were made available by the computer center--the Office of Computing and Information Services (OCIS).

Brown, Gary DeWard. System/370 Job Control Language. New York: John Wiley & Sons, c1977.

This is a clear, well-written book.

DATASET DUMP UTILITY USEP'S GUIDE. Athens: OCIS; [1976].
Available from: OCIS; Guide "DSDUMP".

DSDUMP is useful for managing files. For example, it can be used to dump files from the MUSIC system or from a tape into files on the IBM system.

DATASET PROTECTION ON THE IBM. Athens: OCIS; 1981.
Available from: OCIS; Memo "108".

This memo describes the KLEEN program which will wipe out datasets which are not "protected." Datasets are protected by filing out papers at the OCIS.

FACILITY ACCESS AND SERVICES GUIDE. Athens: OCIS; 1981.
Available from: OCIS; Guide "IBMINFO".

Jahoda, Gerald. Information storage and retrieval systems for individual researchers. New York: Wiley-Interscience, 1970.

An excellent introduction intended to help researchers get their personal files in order.

Klieger, Douglas M. Computer Usage for Social Scientists. Boston: Allyn and Bacon, Inc, c1984.

The first five chapters are an introduction to computing and the remaining chapters discuss the SPSS, SAS, and BMDP statistical packages.

MUSIC General User's Reference Card. 1st ed. White Plains, NY: IBM Corporation; 1981.
Available from: University of Georgia Bookstore.

This card gives the commands needed by the MUSIC system user.

MUSIC TUTORIAL. Athens: OCIS; 1983.
Available from: OCIS Billing Office, Room 122A in the Graduate Studies Building.

Contains useful examples of actual terminal sessions.

MUSIC USER'S GUIDE. Athens: OCIS, 1982.
Available from: OCIS; Guide "MUSIC".

A detailed manual on MUSIC. The user should be aware that some of the features described have not been implemented at the University of Georgia.

Ott, James W. TUCC TSO EDITOR, QED. Release 3. Research Triangle Park, NC: Triangle Universities Computation Center, 1977.
Available from: OCIS; Guide "QED".
The "banner" page gives the title as: TUCC QUICK EDITOR REFERENCE MANUAL.

QED is an interactive editor program. It can be used to create and edit files on the IBM system.

UNIVERSITY OF GEORGIA--IBM 6670--User Reference [DRAFT]. [Athens: OCIS, n.d.].

The IBM 6670 Information Destributor is the "laser printer." This is not available from OCIS yet. The user should contact the User Services Office of the OCIS for information about the laser printer.

WHAT YOU SHOULD KNOW ABOUT JES3 CARDS. Athens: OCIS, 1983.
Available from: OCIS; Memo "217."

D.2 FAMULUS ARTICLES

This is a list of books and articles about FAMULUS and its applications.

Ashley, Tom R. FAMULUS: A reprint classification system for the research scientist. Entomological Society of America Bulletin. 27(3):161-165. 1981.

Batty, C. David; Destricker, U.; Polson, L. Famuletta: A FAMULUS Sampler. Montreal: McGill University, Graduate School of Library Science; 1976.

Batty, C. David; Destricker, U.; Polson, L. Illustrated Guide to FAMULUS. Montreal: McGill University, Graduate School of Library Science; 1976.

Bolin, Robert L. A FAMULUS Index to the Bulletin of the University of Georgia. Athens, GA: University of Georgia, Political Science Department; 1984.
Available in: University of Georgia Libraries, Special Collections Department.

Bolin, Robert L. A FAMULUS Index to Georgia State College of Agriculture Bulletins and Circulars. Athens, GA: University of Georgia, Political Science Department; 1984.

Burton, Hilary D.; Yerke, Theodor B. FAMULUS: A Computer-Baser System . . . Proceedings of the American Society for Information Science. 6: 53-56; 1969. Also available as Forest Service Research Note PSW-193.

Cochran, M.I.; Thomas, J.M. Application of FAMULUS: A Bibliography of Quantitative Ecology. Richland, WA: Battelle Pacific Northwest Labs; 1976.
Available from: NTIS, Springfield, VA; BNWL-2130.

Crennell, K.M. Software Review: FAMULUS. Computers and the Humanities. 10:233-234; 1976.
A good, brief description.

Crennell, K.M.; Marsh, E. Computerisation of the Rutherford Laboratory Catalogue. Program. 13(2): 73-84; 1979 April.

Grimsdale, J. Meteorology: Literature Indexing and Information Retrieval Using the FAMULUS System. Saskatchewan: Saskatchewan Research Council; 1974.

Hockey, Susan. A Guide to Computer Applications in the Humanities. Baltimore, MD: Johns Hopkins University Press; 1981.
Chapter 9: Indexing, Cataloging, and Information Retrieval.

Lengel, Joan E.; Koning, John W., Jr. An Information Retrieval System for Research File Data. Madison, WI: Forest Products Laboratory; 1976.
Available from: NTIS, Springfield, VA; AD-A067 021/5GA.

Martignon, M.E.; Williams, P.; Reineke, D.E. Computer-Based Catalog of Viral Diseases of Insects--FAMULUS Application. Journal of Invertebrate Pathology. 22(1): 100-107; 1973.

McCrank, Lawrence J.; Batty, C. David. The Mt. Angel Abbey Manuscript and Rare Book Project: Cataloging with FAMULUS. Computers and the Humanities. 12(1/2): 215-222; 1978.

Pao, Miranda Lee. File Construction Using FAMULUS. Special Libraries. 73(1): 46-51; 1982 January.
Describes use of FAMULUS to teach indexing.

Scharpf, Robert F.; Hawksworth, Frank G.; Erickson, Bernard J. Mistletoe Literature of the World; A User's Guide to a FAMULUS retrieval system. Ft Collins, CO: Rocky Mountain Forest and Range Experiment Station; 1976.
Forest Service General Technical Report RM-30.

Tucker, Allen B, Jr. Text Processing: Algorithms, Languages, and Applications. New York: Academic Press, 1979.
Chapter 4: Overview of Text Processing Packages and Applications.

Yerke, Theodor B. Computer Support of the Researcher's Own Documentation. Datamation. 16(2): 75-77; 1970.